PICmicro® Microcontroller Firmware Flow Chart of DV103006 Demo Reader for MCRF3XX and MCRF4XX Devices

RFID Top-Level

N = No operation
C = Configuration message
M = Mode select
R = Reset request
W = Write
V = Verbose read
L = Load

WAIT FOR START OF COMMAND
GET COMMAND LETTER
CMD = “N”
CMD = “C”
CMD = “M”
CMD = “R”
CMD = “W”
CMD = “V”
CMD = “L”

DECODE AND CONFIGURE

JUMP TO PROGRAM LOCATION “0000”

PERFORM “READ” COMMAND

EXECUTE SPECIFIED MODE

SEND “READY” MESSAGE

PERFORM “WRITE” COMMAND

JUMP TO LOADER

Loads new firmware using RS-232 port

SEND ERROR MSG:
“UNDEFINED” CMD

FLUSH REST OF CMD LINE
CONFIGURE

READ PARAMETER ID

READ PARAMETER VALUE

ID = "I"

Yes

SET AUDIO ON/OFF

No

ID = "G"

STORE DELTA GAP-WIDTH

ID = "P"

Yes

STORE DELTA GAP PERIOD

No

ID = "M"

STORE TC MAX

ID = "T"

STORE TS

Yes

SEND ERROR MSG: "UNDEFINED" PARAMETER

No

ID = "S"

STORE NORMAL/FAST FLAG

ID = "V"

SET NEW CARRIER VOLTAGE

No

End of Line

A
In case any tags were put to sleep in previous activity

MODE

READ MODE LETTER

NEED TO CYCLE THE CARRIER?

TURN OFF CARRIER

WAIT 100 ms

TURN ON CARRIER

WAIT 250 ms

TURN CARRIER OFF

TURN CARRIER ON

MODE = "F"

MODE = "S"

MODE \in \{0,1,2,3\}

MODE \in \{I,A,C\}

SEND ERROR MSG: "UNKNOWN MODE"
MODES 0, 1, 2, 3

MCRF355

TURN ON CARRIER

CLEAR TAG DATABASE

RECEIVED ANY RS-232?

WAIT FOR TAG DATA

GOT 18 BYTES?

REMOVE '0' BITS BETWEEN EVERY BYTE

ANY SPACE BITS = '1'?

REVERSE BIT ORDER IN EACH BYTE

CALCULATE CHECKSUM

CHECKSUM CORRECT?

MODE = '0' OR '1'?
MODES A, I, C

- **Mode A**: Look for FRR tag only
- **Mode I**: Put tags to sleep when they are found
- **Mode C**: Look for FRR & FRB tags continuously

**MCRF450**

- **RECEIVED ANY RS-232 BYTES?**
  - **Yes**: SEND FOR GAP SEQUENCE
  - **No**: WAIT FOR TAG DATA

- **TIME-OUT?**
  - **Yes**: MODE = 'A'?
    - **Yes**: SEND FRB GAP SEQUENCE
    - **No**: SEND FRR RESPONSE PACKET TO PC
  - **No**: MODE = 'I'?
    - **Yes**: SEND: CALIBRATION & MC1
    - **No**: SEND FRB RESPONSE PACKET TO PC

- **NOTE**: This puts part to Sleep for Inventory mode.

**MODE = 'A'?**

- **Yes**: SEND FRB GAP SEQUENCE
  - **TIME-OUT?**
    - **Yes**: SEND: CALIBRATION + MC1 + END PROCESS
    - **No**: SEND FRB RESPONSE PACKET TO PC
  - **No**: MODE = 'I'?
    - **Yes**: SEND: CALIBRATION + MC1 + END PROCESS
    - **No**: SEND FRB RESPONSE PACKET TO PC

**MODE = 'I'?**

- **Yes**: SEND FRB GAP SEQUENCE
  - **TIME-OUT?**
    - **Yes**: SEND: CALIBRATION + MC1 + END PROCESS
    - **No**: SEND FRB RESPONSE PACKET TO PC
  - **No**: MODE = 'A'?
READ COMMAND

GET TAG ID FROM PC

NEED TO CYCLE THE CARRIER?

TURN OFF CARRIER

WAIT 100 ms

TURN ON CARRIER

WAIT 250 ms

ACCESS A TAG

BLK_NO = 0

READ BLOCK

OK?

REPORT TO PC: BLK_NO: DATA

REPORT TO PC: DATA = "XXXX"

++BLK_NO = 32?

TURN OFF CARRIER

EXIT
WRITE COMMAND

GET TAG_ID, STARTING BLOCK_NO, BLOCK DATA FROM PC

TAG_ID = ""?

Yes → WR_ALL_TAGS = 1

No → BLOCK DATA BIT 31 = 1

Yes → START BLOCK = 0

No → BLOCK DATA BIT 30 = 1

Yes → MAKE_FRR = 1

No → MAKE_FRB = 1

NEED TO CYCLE THE CARRIER?

Yes → CARRIER OFF FOR 100 ms

No → CARRIER ON FOR 250 ms

ACCESS A TAG

MAKE INTO AN FRR PART

Yes → MAKE_FRR?

No → MAKE_FRB?

Yes → START BLOCK = 0,3,4,5?

No → READ BLOCK 2

WRITE BLOCKS

PUT TAG TO SLEEP

Yes → WR_ALL_TAGS?

No → TURN OFF CARRIER → EXIT

START BLOCK = 0,3,4,5?
READ BLOCK

RETRIES = 3

SEND READ BLOCK CMD FOR BLK_NO

WAIT FOR TAG DATA

ANY DATA?

Yes

MESSAGE CRC CORRECT?

Yes

EXIT

(PASS)

No

RETRIES > 0?

Yes

EXIT

(FAIL)

No

EXIT

Yes

No

No

Yes
WRITE BLOCKS

BEGIN RESPONSE PACKET TO PC

MAKE FRB?

MAKE FRR?

IS BLOCK WRITE PROTECTED?

SEND WRITE_BLOCK COMMAND TO TAG

RETRIES = 8

WAIT FOR TAG DATA

MAKE FRR?

TIME-OUT?

REPORT "RO" FOR BLOCK DATA

REPORT "XXXX" FOR BLOCK DATA

SEND READ_BLOCK COMMAND TO TAG

BLOCK_NO = BLOCK_NO + 1

MORE BLOCKS TO WRITE?

SEND SET FR_BIT COMMAND TO TAG

SET-UP TO LOOP ONCE MORE

FINISH RESPONSE PACKET TO PC

EXIT

E

IS THIS AN FRR PART?

MAKE FRR?

ANY ERRORS SO FAR?

IS THIS BLOCK WRITE PROTECTED?

IS BLOCK 0 CRC CORRECT?

IS BLOCK 0 CRC CORRECT?
MAKE AN FRR PART

READ TAG’S BLOCK #0

FAIL? Yes
No

READ TAG’S BLOCK #2

FAIL? Yes
No

IS THIS TAG AN FRR ALREADY?

Yes
No

READ TAG’S BLOCK #’S 3, 4, 5

FAIL? Yes
No

CALCULATE CORRECT CRC FOR FRR RESPONSE

STORE CRC IN BLOCK 0 RAM BUFFER

WRITE BLOCK #0

SEND END PROCESS CMD TO SLEEP THE TAG

WR_ALL_TAGS? Yes
No

TURN CARRIER OFF

EXIT

Wait for command from PC
SLAVE PROCESSOR

INITIALIZE

STATE = 0

WAIT FOR FIRST EDGE OF MANCHESTER DATA

ENABLE INTERRUPTS

WAIT FOR END OF DATA STREAM

DISABLE INTERRUPTS

IS MASTER µP BUSY?
Yes
WAIT FOR MASTER PROCESSOR READY

No
ANY DATA RCVD FROM TAG?
Yes
WAIT FOR MASTER PROCESSOR READY

No
ANY PARTIAL BYTE REMAINING?
Yes
SEND PARTIAL BYTE OVER SPI BUS

No
SEND # EMPTY BITS IN FINAL DATA BYTE OVER SPI BUS
SLAVE PROCESSOR ISR

CAPTURE CURRENT STATE OF MANCHESTER DATA

CAPTURE TIME ELAPSED SINCE LAST EDGE ON MANCHESTER LINE

STATE = 0

STATE = 1

STATE = 2

ADD THE BIT TO BYTE-BUILDING BUFFER

IS THIS THE 4TH EDGE?

Yes

STATE = 1

IS THIS A ZERO BIT?

Yes

STATE = 3

SEND FIRST BYTE (ALWAYS 7F) OVER SPI BUS

8TH BIT?

HAS MASTER TAKEN PREVIOUS BYTE OFF SPI BUS?

Yes

SEND LATEST BYTE TO MASTER VIA SPI BUS

EXIT ISR

FLAG: MASTER µP IS BUSY
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.